

CONTEÚDO

PREFÁCIO

AO LEITOR

A Importância de C

Público-alvo

ESCOPO DO LIVRO

ORGANIZAÇÃO DO LIVRO

Volume I

Volume II

COMO USAR O LIVRO

Recomendações ao Aprendiz

Recomendações ao Instrutor

MATERIAL COMPLEMENTAR

Hardware e Software

Exemplos e Códigos-fonte

CONVENÇÕES ADOTADAS NO LIVRO

Convenções Tipográficas

Outras Convenções

Simplificações

Práticas Inimitáveis

CRÍTICAS, SUGESTÕES E COMENTÁRIOS

AGRADECIMENTOS

CAPÍTULO 1 – INTRODUÇÃO À LINGUAGEM DE PROGRAMAÇÃO C

1.1 HISTÓRICO DA LINGUAGEM C

1.2 ASPECTOS FUNDAMENTAIS

1.2.1 Identificadores e Palavras-chaves

1.2.2 Tipos de Dados Primitivos

1.2.3 Constantes

1.2.4 Definições de Variáveis

1.2.5 Atribuição

1.2.6 Constantes Simbólicas

1.2.7 Comentários

1.3 OPERADORES E EXPRESSÕES

1.3.1 Operadores

1.3.2 Expressões

1.3.3 Propriedades dos Operadores

1.3.4 Uso de Parênteses

1.3.5 Operadores e Expressões Aritméticos

1.3.6 Operadores e Expressões Relacionais

1.3.7 Operadores e Expressões Lógicas

1.4 OS TIPOS _Bool E bool

1.5 MISTURA DE TIPOS E CONVERSÕES AUTOMÁTICAS

1.5.1 Conversão de Atribuição

1.5.2 Conversão de Alargamento (Promoção)

1.5.3 Conversão Aritmética Usual

1.5.4 Problemas Causados por Conversões Automáticas

1.6 OUTROS OPERADORES

1.6.1 Conversão Explícita

1.6.2 O Operador de Atribuição

1.6.3 Operadores de Atribuição Aritmética

1.6.4 Operadores de Incremento e Decremento

1.6.5 O Operador sizeof

1.7 ESTRUTURAS DE CONTROLE

- 1.7.1 Seqüências de Instruções
- 1.7.2 Instruções Vazias
- 1.7.3 Laços de Repetiço: while, do-while e for
- 1.7.4 Desvios Condicionais: if-else e switch-case
- 1.7.5 Desvios Incondicionais: break, continue e goto
- 1.8 O OPERADOR CONDICIONAL
- 1.9 O OPERADOR VRGULA
- 1.10 EXERCCIOS DE REVISO

CAPTULO 2 – PROGRAMAS MONO-ARQUIVOS

- 2.1 INTRODUÇO
- 2.2 INTRODUÇO  BIBLIOTECA PADRO DE C
- 2.3 FUNÇES ELEMENTARES DE ENTRADA E SADA
 - 2.3.1 O Mdulo stdio
 - 2.3.2 Funço getchar()
 - 2.3.3 Funço putchar()
 - 2.3.4 Funço scanf()
 - 2.3.5 Funço printf()
- 2.4 COMO CONSTRUIR UM PROGRAMA SIMPLES EM C
 - 2.4.1 Construço de Programas Pequenos numa Linguagem Algortmica
 - 2.4.2 Estrutura de um Programa Simples em C
 - 2.4.3 Utilizando um Ambiente Integrado de Desenvolvimento
 - 2.4.4 Utilizando Editor de Programas e gcc
- 2.5 EXECUTANDO UM PROGRAMA
- 2.6 USANDO A ENTRADA DE DADOS PADRO
 - 2.6.1 Entendendo Entrada de Dados Padro
 - 2.6.2 Uso de Laços de Repetiço em Leitura de Dados
- 2.7 EXERCCIOS DE REVISO
- 2.8 EXERCCIOS DE PROGRAMAÇO

CAPTULO 3 – FUNÇES

- 3.1 INTRODUÇO
- 3.2 ENDEREÇOS E PONTEIROS
 - 3.2.1 Endereços
 - 3.2.2 Ponteiros
 - 3.2.3 Indireço de Ponteiros
 - 3.2.4 O Ponteiro Nulo
 - 3.2.5 Compatibilidade e Converses entre Ponteiros
- 3.3 FUNÇES
 - 3.3.1 Definiçes de Funçes
 - 3.3.2 Chamadas de Funçes
 - 3.3.3 Passagem de Parmetros
 - 3.3.4 Instruço return
 - 3.3.5 Converses Automticas em Chamada e Retorno de Funçes
 - 3.3.6 Prottipos e Aluses de Funçes
- 3.4 FUNÇES COM LISTAS DE ARGUMENTOS VARIVEIS
- 3.5 RECURSO
 - 3.5.1 Funçes Recursivas
 - 3.5.2 O Problema das Torres de Hani
 - 3.5.3 Cadeias Recursivas
- 3.6 FUNÇES INLINE (C99)
- 3.7 LEITURA E VALIDAÇO DE DADOS I
 - 3.7.1 Sinalizaço do Final do Programa

- 3.7.2 Leitura de Dados com Validação
- 3.7.3 Teste dos Valores Lidos
- 3.7.4 Limpeza do Buffer de Entrada
- 3.7.5 Chamada da Função de Leitura
- 3.7.6 Limitações do Método Utilizado
- 3.8 INTERAÇÃO DIRIGIDA POR MENUS
- 3.9 EXERCÍCIOS DE REVISÃO
- 3.10 EXERCÍCIOS DE PROGRAMAÇÃO

CAPÍTULO 4 – PROGRAMAS MULTIARQUIVOS

- 4.1 INTRODUÇÃO
- 4.2 DURAÇÃO DE VARIÁVEIS
 - 4.2.1 Duração Fixa e Duração Automática de Variáveis
 - 4.2.2 Iniciação de Variáveis de Acordo com a Duração
- 4.3 ESCOPO
 - 4.3.1 Escopo de Programa
 - 4.3.2 Escopo de Arquivo
 - 4.3.3 Escopo de Função
 - 4.3.4 Escopo de Bloco
 - 4.3.5 Hierarquia de Escopos
 - 4.3.6 Conflitos de Identificadores
- 4.4 ESPECIFICADORES DE REGISTRADORES
- 4.5 QUALIFICADORES DE TIPOS
 - 4.5.1 Qualificador const
 - 4.5.2 Qualificador volatile
- 4.6 TIPOS DE DADOS DEFINIDOS PELO PROGRAMADOR
- 4.7 VARIÁVEIS GLOBAIS
 - 4.7.1 Recomendações de Uso
 - 4.7.2 Definições e Alusões
- 4.8 FUNÇÕES GLOBAIS E FUNÇÕES DE ARQUIVO
- 4.9 LIGAÇÕES DE IDENTIFICADORES
- 4.10 MÓDULOS
- 4.11 COMO CONSTRUIR PROGRAMAS MULTIARQUIVOS
 - 4.11.1 Modelos de Arquivos para Programas Multiarquivos
 - 4.11.2 Utilizando um Ambiente Integrado de Desenvolvimento
 - 4.11.3 Utilizando Editor de Texto e gcc
 - 4.11.4 Make e Arquivos Makefiles
- 4.12 EXERCÍCIOS DE REVISÃO
- 4.13 EXERCÍCIOS DE PROGRAMAÇÃO

CAPÍTULO 5 – O PRÉ-PROCESSADOR DE C

- 5.1 INTRODUÇÃO
- 5.2 UNIDADES DE TRADUÇÃO
- 5.3 MACROS
 - 5.3.1 Macros sem Argumentos
 - 5.3.2 Macros com Argumentos
 - 5.3.3 Comparações entre Macros com Argumentos e Funções
 - 5.3.4 Comparações entre Macros com Argumentos e Funções Inline (C99)
 - 5.3.5 Macros Produtoras de Strings: O Operador #
 - 5.3.6 Concatenação de Tokens: O Operador ##
 - 5.3.7 Removendo a Definição de uma Macro
 - 5.3.8 Macros Predefinidas
 - 5.3.9 Macros Vazias

- 5.3.10 Macros com Listas de Argumentos Variáveis (C99)
- 5.3.11 Cuidados com Usos de Macros
- 5.4 COMPILAÇÃO CONDICIONAL
 - 5.4.1 Testando o Valor de Macros: Diretivas #if, #else, #elif e #endif
 - 5.4.2 Testando a Existência de Macros: Diretivas #ifdef e #ifndef
- 5.5 INCLUSÃO DE ARQUIVOS
- 5.6 A DIRETIVA #error
- 5.7 A DIRETIVA #pragma
- 5.8 O OPERADOR _Pragma (C99)
- 5.9 A DIRETIVA #line
- 5.10 DIRETIVA NULA
- 5.11 EXERCÍCIOS DE REVISÃO
- 5.12 EXERCÍCIOS DE PROGRAMAÇÃO

CAPÍTULO 6 – LEGIBILIDADE E DEPURAÇÃO DE PROGRAMAS

- 6.1 INTRODUÇÃO
- 6.2 USO DE COMENTÁRIOS
 - 6.2.1 Comentários de Bloco
 - 6.2.2 Comentários de Linha
- 6.3 USO DE ESPAÇOS EM BRANCO
 - 6.3.1 Endentação
 - 6.3.2 Outros Espaços em Branco Horizontais
 - 6.3.3 Espaços em Branco Verticais
- 6.4 ESCOLHA DE IDENTIFICADORES
 - 6.4.1 Convenções
 - 6.4.2 Representatividade de Identificadores
- 6.5 RECOMENDAÇÕES PARA ESCRITA DE INSTRUÇÕES E EXPRESSÕES
- 6.6 NÚMEROS MÁGICOS
- 6.7 USO DE JARGÕES
- 6.8 INTERFACE DO USUÁRIO
- 6.9 GUIAS DE ESTILO DE PROGRAMAÇÃO EM C
- 6.10 TESTE E DEPURAÇÃO DE PROGRAMAS
 - 6.10.1 Testando um Programa
 - 6.10.2 Depuração de Programas
 - 6.10.3 Classificação de Erros de Programação
 - 6.10.4 Poucando Depuração
- 6.11 TÉCNICAS ELEMENTARES DE DEPURAÇÃO
 - 6.11.1 Uso do Compilador e de Lint
 - 6.11.2 Uso de printf()
 - 6.11.3 Uso de Comentários
 - 6.11.4 Uso de Asserções e Compilação Condicional
- 6.12 DEPURADORES DE ALTO NÍVEL
- 6.13 DEPURAÇÃO USANDO GDB
 - 6.13.1 Preparação
 - 6.13.2 Executando gdb
 - 6.13.3 Obtendo Ajuda sobre Comandos
 - 6.13.4 Executando um Programa
 - 6.13.5 Situando-se com Backtrace e List
 - 6.13.6 Encerrando a Execução de um Programa
 - 6.13.7 Corrigindo um Erro
 - 6.13.8 Depuração Post-mortem: Examinando um Arquivo Core
 - 6.13.9 Instalando e Removendo Breakpoints
 - 6.13.10 Examinando Valores de Variáveis: Comando Print

- 6.13.11 Controlando a Execução de um Programa
- 6.13.12 Selecionando um Contexto: Comando Frame
- 6.13.13 Examinando Valores de Variáveis: Comando Display
- 6.13.14 Outros Comando do gdb
- 6.13.15 O Depurador gdb como Ferramenta de Aprendizagem
- 6.14 DEPURAÇÃO USANDO DDD
- 6.15 OTIMIZAÇÃO DE CÓDIGO
- 6.16 EXERCÍCIOS DE REVISÃO
- 6.17 EXERCÍCIOS DE PROGRAMAÇÃO

CAPÍTULO 7 – ARRAYS E PONTEIROS

7.1 INTRODUÇÃO

7.2 ARRAYS

- 7.2.1 Definições de Arrays
- 7.2.2 Acessando Elementos de um Array
- 7.2.3 Iniciações de Arrays
- 7.2.4 Inicializadores Designados de Arrays (C99)

7.3 ARITMÉTICA DE PONTEIROS

7.4 RELAÇÃO ENTRE PONTEIROS E ARRAYS UNIDIMENSIONAIS

7.5 ARRAYS COMO PARÂMETROS DE FUNÇÕES

- 7.5.1 Declarando Arrays como Parâmetros Formais
- 7.5.2 Arrays como Parâmetros Reais
- 7.5.3 Arrays com Índices Estáticos e Qualificadores de Tipos (C99)
- 7.5.4 Retorno de Arrays

7.6 ARRAYS MULTIDIMENSIONAIS

- 7.6.1 Iniciações de Arrays Multidimensionais
- 7.6.2 Acesso a Elementos de Arrays Multidimensionais
- 7.6.3 Relação entre Ponteiros e Arrays Multidimensionais
- 7.6.4 Arrays Multidimensionais como Parâmetros de Funções

7.7 ARRAYS DE TAMANHOS VARIÁVEIS (C99)

7.8 PONTEIROS RESTRITOS (C99)

7.9 IMITAÇÕES DE PONTEIROS (C99)

7.10 EXERCÍCIOS DE REVISÃO

7.11 EXERCÍCIOS DE PROGRAMAÇÃO

CAPÍTULO 8 – STRINGS

8.1 INTRODUÇÃO

8.2 DEFINIÇÕES E INICIAÇÕES DE STRINGS

8.3 COMPARAÇÃO ENTRE PONTEIROS, STRINGS E CARACTERES

8.4 FUNÇÕES DE BIBLIOTECA PARA PROCESSAMENTO DE STRINGS

- 8.4.1 Entrada de Strings
- 8.4.2 Saída de Strings
- 8.4.3 Comprimento de Strings: Função strlen()
- 8.4.4 Cópia de Strings: Função strcpy()
- 8.4.5 Concatenação de Strings: Função strcat()
- 8.4.6 Casamento de Strings: Função strstr()
- 8.4.7 Comparação de Strings: Função strcmp()
- 8.4.8 Procurando um Caractere num String: Funções strchr() e strrchr()
- 8.4.9 Procurando Caracteres num String: Funções strpbrk(), strspn() e strcspn()
- 8.4.10 Separando Strings em Tokens: Função strtok()
- 8.4.11 Outras Funções para Processamento de Strings

8.5 A FUNÇÃO main()

8.6 O IDENTIFICADOR PREDEFINIDO __func__ (C99)

- 8.7 CARACTERES UNIVERSAIS (C99)
- 8.8 LEITURA SEGURA DE STRINGS
 - 8.8.1 Problemas com as Funções de Leitura de Strings da Biblioteca Padrão
 - 8.8.2 Algoritmo para Leitura de Strings
 - 8.8.3 Implementação da Função de Leitura de Strings
 - 8.8.4 Limpeza do Buffer de Entrada
 - 8.8.5 Uso da Função de Leitura
- 8.9 EXERCÍCIOS DE REVISÃO
- 8.10 EXERCÍCIOS DE PROGRAMAÇÃO

CAPÍTULO 9 – ESTRUTURAS, UNIÕES E ENUMERAÇÕES

- 9.1 INTRODUÇÃO
- 9.2 DEFINIÇÕES E INICIAÇÕES DE ESTRUTURAS
 - 9.2.1 Definições de Estruturas Usando Declarações de Rótulos
 - 9.2.2 Definições de Estruturas Usando Definições de Tipos
 - 9.2.3 Iniciações de Estruturas
 - 9.2.4 Atribuição entre Estruturas
- 9.3 ACESSO A CAMPOS DE ESTRUTURAS
- 9.4 ESTRUTURAS ANINHADAS
- 9.5 ESTRUTURAS COM AUTO-REFERÊNCIA
- 9.6 ESTRUTURAS COMO PARÂMETROS E RETORNOS DE FUNÇÕES
 - 9.6.1 Estruturas como Parâmetros de Funções
 - 9.6.2 Funções com Retorno de Estruturas
- 9.7 ARRAYS DE ESTRUTURAS
- 9.8 ARRAYS FLEXÍVEIS COMO MEMBROS DE ESTRUTURAS (C99)
- 9.9 UNIÕES
- 9.10 INICIADORES DESIGNADOS DE ESTRUTURAS E UNIÕES (C99)
- 9.11 LITERAIS COMPOSTOS (C99)
- 9.12 ENUMERAÇÕES
- 9.13 TIPOS INCOMPLETOS E COMPATIBILIDADE DE RÓTULOS
- 9.14 EXERCÍCIOS DE REVISÃO
- 9.15 EXERCÍCIOS DE PROGRAMAÇÃO

CAPÍTULO 10 – CONSTRUÇÕES COMPLEXAS

- 10.1 INTRODUÇÃO
- 10.2 OPERADORES E DECLARADORES
 - 10.2.1 Operadores de Memória
 - 10.2.2 Declaradores
- 10.3 ARRAYS DE PONTEIROS
- 10.4 PONTEIROS PARA PONTEIROS
- 10.5 PONTEIROS PARA FUNÇÕES
 - 10.5.1 Definições de Ponteiros para Funções
 - 10.5.2 Atribuição de Valores a um Ponteiro para Função
 - 10.5.3 Chamada de Função Mediante Ponteiro
 - 10.5.4 Retornando Ponteiros para Funções
 - 10.5.5 Ponteiros para Funções como Parâmetros de Funções
- 10.6 INTERPRETANDO CONSTRUÇÕES COMPLEXAS
- 10.7 COMENDO CONSTRUÇÕES COMPLEXAS
- 10.8 ALGORITMOS DE CLASSIFICAÇÃO I
 - 10.8.1 O Método da Bolha
 - 10.8.2 Generalizando a Função BubbleSort() I
- 10.9 LEITURA E VALIDAÇÃO DE DADOS II
 - 10.9.1 Descrição Geral do Método

- 10.9.2 Funções, Variáveis e Tipos da Biblioteca Padrão Utilizados
- 10.9.3 Leitura e Validação de Valores
- 10.9.4 Apresentação de Mensagens de Erro
- 10.9.5 Indicação Precisa de Erros
- 10.9.6 Função LimpaBuffer()
- 10.9.8 Chamada da Função de Leitura
- 10.9.9 Exemplo de Interação
- 10.9.10 Limitações da Implementação
- 10.10 EXERCÍCIOS DE REVISÃO
- 10.11 EXERCÍCIOS DE PROGRAMAÇÃO

CAPÍTULO 11 – ALOCAÇÃO DINÂMICA DE MEMÓRIA

- 11.1 INTRODUÇÃO
- 11.2 FUNÇÕES DE ALOCAÇÃO DINÂMICA DE MEMÓRIA
 - 11.2.1 A Função malloc()
 - 11.2.2 A Função calloc()
 - 11.2.3 A Função realloc()
 - 11.2.4 A Função free()
- 11.3 O HEAP
- 11.4 PONTEIROS GENÉRICOS
- 11.5 EXEMPLO DE ALOCAÇÃO DINÂMICA DE MEMÓRIA
- 11.6 ESTRUTURAS RECURSIVAS E LISTAS ENCADEADAS
- 11.7 IMPLEMENTAÇÃO DE LISTAS ENCADEADAS EM C
- 11.8 OPERAÇÕES BÁSICAS SOBRE LISTAS ENCADEADAS
 - 11.8.1 Criação de uma Lista Encadeada
 - 11.8.2 Visita Seqüencial aos Elementos de uma Lista Encadeada
 - 11.8.3 Inserção de Elementos numa Lista Encadeada
 - 11.8.4 Remoção de Elementos de uma Lista Encadeada
- 11.9 ALGORITMOS DE CLASSIFICAÇÃO II
 - 11.9.1 A Função de Biblioteca qsort()
 - 11.9.2 Generalizando a Função BubbleSort() II
- 11.10 EXERCÍCIOS DE REVISÃO
- 11.11 EXERCÍCIOS DE PROGRAMAÇÃO

CAPÍTULO 12 – PROCESSAMENTO DE ARQUIVOS

- 12.1 INTRODUÇÃO
- 12.2 STREAMS
 - 12.2.1 O Conceito de Stream
 - 12.2.2 Implementação de Streams em C: Estruturas do Tipo FILE
- 12.3 BUFFERING
- 12.4 FORMATOS DE ARQUIVOS
 - 12.4.1 Arquivos de Texto
 - 12.4.2 Arquivos Binários
 - 12.4.3 Comparação entre Arquivos de Texto e Arquivos Binários
- 12.5 ABRINDO E FECHANDO UM ARQUIVO
 - 12.5.1 Abrindo um Arquivo: Função fopen()
 - 12.5.2 Fechando um Arquivo: Função fclose()
- 12.6 STREAMS PADRÃO: stdin, stdout E stderr
- 12.7 REDIRECIONAMENTO DE ENTRADA E SAÍDA PADRÃO
 - 12.7.1 Redirecionamento para Arquivos de Texto
 - 12.7.2 Redirecionamento para Impressora
 - 12.7.3 Redirecionamento Utilizando a Função freopen()
- 12.8 PROCESSAMENTO SEQÜENCIAL DE ARQUIVOS

12.8.1 Processando Caracteres: Funções fgetc(), fputc(), getc() e putc()

12.8.2 Processando Linhas: Funções fgets() e fputs()

12.8.3 Processando Blocos: Funções fread() e fwrite()

12.9 ACESSO DIRETO

12.10 OUTRAS FUNÇÕES DE ENTRADA E SAÍDA

12.10.1 Entrada Formatada: A Família scanf

12.10.2 Saída Formatada: A Família printf

12.10.3 Formatação em Memória

12.10.4 Especificação de Buffering: Funções setbuf() e setvbuf()

12.10.5 Gerenciamento de Arquivos: Funções remove() e rename()

12.10.6 Usando Arquivos Temporários: Funções tmpfile() e tmpnam()

12.10.7 Detecção de Erros em Streams: Funções clearerr() e ferror()

12.10.8 Funções perror() e ungetc()

12.10.9 Funções de Entrada e Saída Não-portáteis

12.11 CLASSIFICAÇÃO EXTERNA DE ARQUIVOS POR INDEXAÇÃO

12.12 EXERCÍCIOS DE REVISÃO

12.13 EXERCÍCIOS DE PROGRAMAÇÃO

CAPÍTULO 13 – PROGRAMAÇÃO DE BAIXO NÍVEL EM C

13.1 INTRODUÇÃO

13.2 SISTEMAS DE NUMERAÇÃO PARA MANIPULAÇÃO DE BITS

13.3 OPERADORES DE MANIPULAÇÃO DE BITS

13.3.1 Operadores Lógicos sobre Bits

13.3.2 Operadores de Deslocamento

13.3.3 Operadores de Atribuição sobre Bits

13.4 MASCARAMENTO

13.5 REPRESENTAÇÃO INTERNA DE VALORES INTEIROS

13.6 CAMPOS DE BITS

13.7 APLICAÇÕES PRÁTICAS

13.7.1 Sinalizadores

13.7.2 Criptografia XOR

13.7.3 Endereçamento IP

13.8 EXERCÍCIOS DE REVISÃO

13.9 EXERCÍCIOS DE PROGRAMAÇÃO

APÊNDICE A – PRECEDÊNCIA E ASSOCIATIVIDADE DE OPERADORES

APÊNDICE B – RESUMO DE ESPECIFICADORES DE FORMATO DE ENTRADA E SAÍDA

B.1 INTRODUÇÃO

B.2 RESUMO DE ESPECIFICADORES DE FORMATO DA FAMÍLIA PRINTF

B.3 RESUMO DE ESPECIFICADORES DE FORMATO DA FAMÍLIA SCANF

BIBLIOGRAFIA

ÍNDICE

LISTA DE FIGURAS

- Figura 1: Tipos de Dados da Linguagem C
- Figura 2: Hierarquia de Tipos Aritméticos
- Figura 3: Processo Simplificado de Construção de um Programa Executável
- Figura 4: Janela Principal do Ambiente DevC++
- Figura 5: Caixa de Configuração do Ambiente DevC++
- Figura 6: Painel de Edição do Ambiente DevC++
- Figura 7: Caixa do DevC++ Indicando Compilação Bem Sucedida
- Figura 8: Mensagem de Advertência Apresentada pelo Ambiente DevC++
- Figura 9: Mensagem de Erro Apresentada pelo Ambiente DevC++
- Figura 10: Habilitação de Numeração de Linhas no Ambiente DevC++
- Figura 11: Execução de um Programa no Sistema DOS/Windows
- Figura 12: Problema das Torres de Hanói
- Figura 13: Hierarquia de Escopos
- Figura 14: Edição de Ligações de um Programa Multiarquivo
- Figura 15: Criando um Projeto Multiarquivo no Ambiente DevC++
- Figura 16: Acrescentando Arquivos a um Projeto no Ambiente DevC++
- Figura 17: Árvore de Projeto Multiarquivo no Ambiente DevC++
- Figura 18: Opções de Compilação e Ligação no Ambiente DevC++
- Figura 19: Processo Completo de Compilação de um Arquivo-fonte
- Figura 20: Processo Completo de Construção de um Arquivo Executável
- Figura 21: Inclusão Múltipla de um Arquivo de Cabeçalho
- Figura 22: Interface ddd em Uso
- Figura 23: Janela de Comandos do ddd
- Figura 24: Soma de um Inteiro a um Ponteiro
- Figura 25: Subtração de um Inteiro a um Ponteiro
- Figura 26: Exemplo de Uso Incorreto do Qualificador restrict
- Figura 27: Funcionamento da Função strtok()
- Figura 28: Função realloc(): Novo Bloco É Menor do que o Bloco Original
- Figura 29: Função realloc(): Novo Bloco É Maior do que o Bloco Original
- Figura 30: Diagrama de uma Lista Simplesmente Encadeada com Três Nós
- Figura 31: Exemplo de Lista Encadeada Contendo Strings
- Figura 32: Criação do Primeiro Nó de uma Lista Encadeada
- Figura 33: Preenchimento do Primeiro Nó de uma Lista Encadeada
- Figura 34: Criação do Segundo Nó de uma Lista Encadeada
- Figura 35: Preenchimento do Segundo Nó de uma Lista Encadeada
- Figura 36: Criação do Terceiro Nó de uma Lista Encadeada
- Figura 37: Preenchimento do Terceiro Nó de uma Lista Encadeada
- Figura 38: Inserção de um Nó numa Lista Encadeada
- Figura 39: Remoção de um Nó de uma Lista Encadeada
- Figura 40: Arquivo de Texto: Interpretação de Conteúdo
- Figura 41: Arquivo Binário: Nenhuma Interpretação de Conteúdo
- Figura 42: Operação de Mascaramento I
- Figura 43: Operação de Mascaramento II
- Figura 44: Operação de Mascaramento III
- Figura 45: Testando se um Sinalizador Está Ligado
- Figura 46: Ligando um Sinalizador
- Figura 47: Desligando um Sinalizador
- Figura 48: Invertendo um Sinalizador

LISTA DE TABELAS

- Tabela 1: Marcos Evolutivos da Linguagem C
- Tabela 2: Intervalos Típicos de Valores de Tipos Inteiros
- Tabela 3: Tipos Complexos Introduzidos pelo Padrão C99
- Tabela 4: Sufixos Utilizados com Constantes Inteiras
- Tabela 5: Regras para Acomodação de Constantes Inteiras (C99)
- Tabela 6: Exemplos de Constantes de Ponto-flutuante em Formato Hexadecimal
- Tabela 7: Seqüências de Escape e seus Significados
- Tabela 8: Operadores Aritméticos
- Tabela 9: Precedências de Operadores Aritméticos
- Tabela 10: Operadores Relacionais
- Tabela 11: Precedências dos Operadores Relacionais
- Tabela 12: Operadores Lógicos em Ordem Decrescente de Precedência
- Tabela 13: Resultados do Operador !
- Tabela 14: Resultados dos Operadores && e ||
- Tabela 15: Precedências Relativas entre Operadores Aritméticos, Relacionais e Lógicos
- Tabela 16: Operadores de Atribuição Aritmética
- Tabela 17: Operadores de Incremento e Decremento
- Tabela 18: Especificadores de Formato Comuns Utilizados por scanf()
- Tabela 19: Especificadores de Formato Comuns Utilizados por printf()
- Tabela 20: Iniciação de Variáveis de Duração Automática e Fixa (Comparação)
- Tabela 21: Definição x Alusão de Variáveis Globais
- Tabela 22: Objetos e Ligações Externos e Internos
- Tabela 23: Comparação entre Escopo e Ligação
- Tabela 24: Diferenças entre Arquivo de Cabeçalho e Arquivo de Programa
- Tabela 25: Modelos de Arquivos e seu Significados
- Tabela 26: Macros Predefinidas
- Tabela 27: Novas Macros Predefinidas Introduzidas pelo Padrão C99
- Tabela 28: Principais Comandos de Execução do Depurador gdb
- Tabela 29: Operações Aritméticas sobre Ponteiros
- Tabela 30: Outras Funções para Processamento de Strings
- Tabela 31: Formatos de Caracteres Universais
- Tabela 32: Operadores de Memória
- Tabela 33: Operadores e Declaradores Correspondentes
- Tabela 34: Precedência e Associatividade de Declaradores
- Tabela 35: Funções de Alocação Dinâmica de Memória
- Tabela 36: Valores Retornados pela Função de Comparação Usada com qsort()
- Tabela 37: Comparação entre Arquivos de Texto e Binários
- Tabela 38: Modos de Acesso para Arquivos de Texto
- Tabela 39: Modos de Acesso para Arquivos Binários
- Tabela 40: Funções para Processamento de Caracteres
- Tabela 41: Protótipos de Funções para Processamento de Caracteres
- Tabela 42: Funções de Posicionamento Utilizadas em Acesso Direto
- Tabela 43: Macros de Posição em Arquivos
- Tabela 44: Funções de Entrada/Saída Padrão e Funções de Streams Genéricos
- Tabela 45: Família de Funções scanf
- Tabela 46: Prefixos Utilizados pela Família de Funções scanf
- Tabela 47: Argumentos Utilizados pela Família de Funções scanf
- Tabela 48: Família de Funções printf
- Tabela 49: Prefixos Utilizados pela Família de Funções printf
- Tabela 50: Argumentos Utilizados pela Família de Funções printf

- Tabela 51: Funções Utilizadas em Formatação em Memória
- Tabela 52: Macros de Especificação de Buffering da Função setvbuf()
- Tabela 53: Macros que Especificam Limites para Arquivos Temporários
- Tabela 54: Conversões entre Bases Numéricas
- Tabela 55: Formatos de Constantes Inteiras Usados em Programação de Baixo Nível
- Tabela 56: Operadores Lógicos Binários sobre Bits
- Tabela 57: Resultados de Operações Lógicas Binárias sobre Bits
- Tabela 58: Operadores Lógicos x Operadores Lógicos sobre Bits
- Tabela 59: Operadores de Deslocamento
- Tabela 60: Operadores de Atribuição Bit a Bit
- Tabela 61: Endereçamento IP: Classes de Endereços
- Tabela 62: Endereçamento IP: Máscaras Padrão das Classes A, B e C
- Tabela 63: Precedência e Associatividade de Operadores da Linguagem C
- Tabela 64: Especificadores de Formato Usados pela Família printf
- Tabela 65: Especificadores de Formato Usados pela Família scanf